

基于遗传算法求解实际车辆调度问题

项光特, 张德富

(厦门大学 计算机科学系, 福建 厦门 361005)

摘要: 车辆路由问题有许多变种, 传统的算法在某些特殊的问题上无法很好的表现。该文提出了一种新型的遗传算法求解思路, 实验表明该方法对带时间窗口多路程的接送车辆路由问题有很好的适应性。

关键词: 带时间窗口多路程车辆路由问题; 遗传算法

中图分类号: TP18 **文献标识码:** A **文章编号:** 1009-3044(2012)11-2595-04

车辆路由问题(Vehicle Routing Problems, VRP)是一个在物流运输及资源调配等方面具有广泛应用的组合优化问题, 是旅行商问题(Traveling Salesman Problem, TSP)的进一步扩展。这个问题可简单描述为: 已知仓库与 n 个客户的地点, 使用 k 辆车满足每个客户的不同需求, 问题的目标是满足客户的需求并使总成本最低。根据实际需求的不同, 考虑不同的约束条件, 可得到大量不同的变种, 例如:

- 1) 带容量限制的车辆路由问题(Capacitated VRP), 每辆车都有最大载货量的限制, 任何时候不能超载;
- 2) 带时间窗口限制的车辆路由问题(VRP with Time Windows), 每个客户必须在某个时间段访问;
- 3) 多仓库的车辆路由问题(Multiple Depot VRP), 仓库点不只一个;
- 4) 带接送的车辆路由问题(VRP with Pick-Up and Delivering), 车辆需要从某地接起货物并运送到指定位置。

该文的第二节描述研究问题; 第三节介绍算法; 第四节分析实验结果; 第五节总结全文。

1 问题描述

多路程的 MT-PDPTW(Multi-Trip Pick-Up and Delivering with Time Windows)是一个某地医院的实际需求中抽象得来的问题。问题本身如下描述: 医院在每天能得知第二天所有需要服务的病人, 每个病人需要从医院用车从某地接送送到另外一个地方, 已知当天能够提供服务的车辆数目及每辆车的具体工作时间, 目标是尽可能多的满足接送病人的数目, 与此同时使路上的费用最短。问题约束条件如下:

- 1) 每个病人都有一个时间窗口, 及必须在这个时间段内去接送, 可以提早到达进行等待;
- 2) 每辆车在外时间最多不能超过2个半小时, 且在完成接送任务后必须回到医院另要进行半个小时的消毒时间, 之后这辆车可以继续为其他病人服务;
- 3) 车辆可以同时接送多个病人, 但是同一时间点上车上的病人数目不能超过车辆的座椅数目(即容量限制);
- 4) 每辆车在中午必须回到医院进行1个小时的午餐时间。

根据上述约束条件, 我们可以建立如下数学模型:

给定车辆集合 K 和无向图 $G=(V, E)$, $V=\{0, 1, \dots, 2n\}$ 是节点, 节点0表示仓库, 节点 $1-2n$ 表示需要接和送的点, 每个点都有一个请求大小 q_i 和服务时间 s_i 。每条路径 (i, j) 都有一个长度 d_{ij} 与旅行时间 t_{ij} 。 $H=\{1, 2, \dots, n\}$ 表示每个接送的请求, p_h 和 d_h 分别是请求 h 的接与送的点, 同时 $q_{ph} = -q_{dh}$ 。 $K=\{1, 2, \dots, m\}$ 表示可供使用的车, 每辆车都有一个属性为最大载客量 Q_k 和车子可供服务的时间段 $[s_k, e_k]$ 。

2 算法设计

遗传算法是一种借鉴进化生物用于解决组合优化问题的搜索算法。进化从随机的个体种群开始, 使用杂交和突变产生新的个体, 然后通过计算适应度来选择产生下一代种群, 直至收敛或某个特定条件被满足。遗传算法的优点在于子代可以通过迭代保留父代的某些优秀遗传因素并累积到后代。

遗传算法的大体框架如下:

- 1) 构造初始解
- 2) 随机选取部分解杂交出下一代
- 3) 一定概率和方法进行变异
- 4) 对子代的解进行选择, 保持种群数目基本不变

2.1 解的表示

根据问题描述, 令 $Trip_i \rightarrow (n_1, n_2, n_3, \dots, n_k)$ 表示一辆车出去至回到医院消毒这一趟行程。令 $TW(Trip)$ 表示 $Trip$ 的时间窗口, $TW(node)$ 表示 $node$ 的时间窗口, $TS(Trip)$ 表示 TS 完成需要的时间, $TS(node)$ 表示 $node$ 需要的服务时间, $Travel_Time(n_1, n_2)$ 表示 $n_1 \rightarrow n_2$ 需

要的时间。

定义操作:

$TimeWindow[a,b] \pm int(d) = [a \pm d, b \pm d]$

$TimeWindow[a,b] \cap TimeWindow[c,d] :$

If $b < c$ then

result is $[b,b]$

Else

result is $[\max(a,c), \min(b,d)]$

$TW(Trip_i) = TW(n_i)$

$TS(Trip_i) = 0 + TS(n_i)$

$TW(Trip_{i-1})$ 可以用下面的方法来得到:

当新增加一个节点 n_i 时,到达该节点的时间窗口为: $TW(Trip_{i-1}) + TS(Trip_{i-1}) + Travel_Time(n_{i-1},n_i)$,

此时我们可以把前面得到时间窗口和节点本身的时间窗口做交集 $TW(n_i)$,

即 $TW(Trip_i) = (TW(Trip_{i-1}) + TS(Trip_{i-1}) + Travel_Time(n_{i-1},n_i)) \cap TW(n_i) - TS(Trip_{i-1}) - Travel_Time(n_{i-1},n_i)$

化简得: $TW(Trip_i) = TW(Trip_{i-1}) \cap (TW(n_i) - TS(Trip_{i-1}) - Travel_Time(n_{i-1},n_i))$

$TS(Trip_i) = \max(TS(Trip_{i-1}) + Travel_Time(n_{i-1},n_i), TW(n_i).start_time) + TS(n_i)$

每个解包含 k 辆车,每辆车包含 t_k 个小回路。根据每个小回路上面每个点的信息我们可以计算出这次回路总共需要的时间和费用及最早和最晚出发的时间窗口。

如某一趟车需要经过4个点 a, b, c, d ,假设车辆在每两个相邻的点需要的时间是10,每个点需要的服务时间是5,每个点的时间窗口如下:

$a: [0, 100]$ $b: [0, 50]$ $c: [40, 60]$ $d: [80, 100]$

我们可以用递推的方法计算出这趟路程的时间及出发时间窗口。

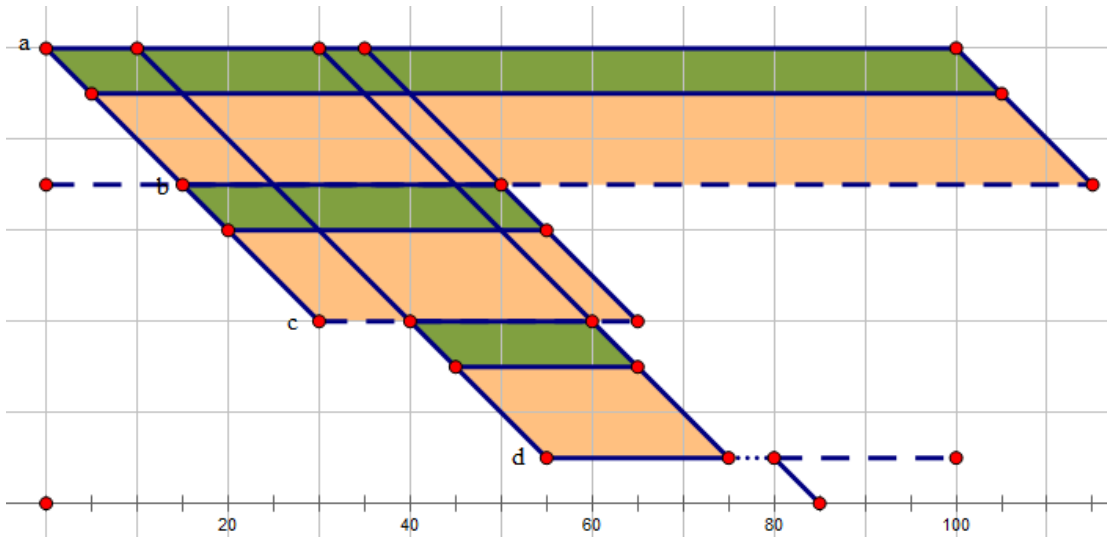


图1 递推求解时间窗口的方法

由图1所示一共需要85时间单位。

这样判断一个解是否合法就是对每辆车判断 n 个 Trip 能否安排在这辆车上。

设 V 是车辆, $t_1, t_2, t_3, \dots, t_n$ 是目前需要安排的 n 个 trip, S 是 $\{t_1, t_2, t_3, \dots, t_n\}$ 。

对于任意 S 的一个子集 s 有:

设 $F(s)$ 表示做完 s 中所有 trip 的最早结束时间,此时我们可以使用动态规划的方法来计算 $F(s)$,算法如下:

- 1 if $s = \{\}$ then $F(s) \leftarrow V.starttime$
- 2 else
- 3 for_each trip t in s
- 4 $s' = s - t$;
- 5 if $F(s') \leq TW(t).endtime$ then
- 6 $F(s) = \min(F(s), \max(F(s'), TW(t).starttime) + TS(t))$

2.2 构造初始解

采用随机顺序插入的方法来构造初始解,算法如下:

- 1 $S \leftarrow \varnothing$
- 2 随机选取一个不在解中的请求 r

```
3  For each single-trip st in S
4  if r 可以安排在这个st里 then
5    S' ← S 删除 st
6    将请求 r 插入在这个st内部并找到最好的位置
7    如果这个st可以安排在当前S'中的某辆车上
8      将 st 安排在 S' 上
9    S ← S'
10 else
11  新建一个新的回路 r ← (0, pr, dr, 0)
12  S' ← S
13  if 这个回路可以安排在当前S'中的某辆车上
14    将 st 安排在 S' 上
15    S ← S'
16 else
17   将 r 放入 S 的未满足集合中
18 return S
```

2.3 染色体定义与杂交

与传统TSP问题不同,另由于问题本身每个小回路路程较短。为了让父代的信息更多的传递到子代,我们把每个小回路看作一个单位。

随机从当前种群中选择2个不同的解P1, P2作为父代,例如P1 为{[0, 1, 2, -1, -2, 0], [0, 3, 4, -4, -3, 0]}, {[0, 6, 5, -6, 7, -5, -7], [0, 8, -8, 0]}

P2 为{[0, 1, 3, -1, -3, 0], [0, 2, 5, -2, -5, 0]}, {[0, 4, -4, 0], [0, 8, 6, -6, -8, 7, -7, 0]}

随机选取P1 中x 个小回路,同时删除P2 中已经被选择的点

如上例,我们随机选择到了 [0, 3, 4, -4, -3, 0] 与[0, 8, -8, 0],删除对应被选择的点后 P2 剩余

{[0, 1, -1, 0], [0, 2, 5, -2, -5, 0]}, {[0, 0], [0, 6, -6, 7, -7, 0]}

[0,0]是没有意义的所以可以删除

{[0, 1, -1, 0], [0, 2, 5, -2, -5, 0]}, {[0, 6, -6, 7, -7, 0]}

此时我们得到了 C'={ [0, 3, 4, -4, -3, 0], [0, 8, -8, 0], [0, 1, -1, 0], [0, 2, 5, -2, -5, 0], [0, 6, -6, 7, -7, 0]}如此5 个小回路。此时我们需要将它们分别安排到对应的车辆上。

算法实现如下:

```
1  S ← φ
2  for each trip t in C'
3    if t 能安排在S 的某辆车上
4    then 将t 安排在该车上
5    else 对 t 中的每一个请求 r
6      尝试将 r 插入到S 中,若失败则将 r 放入 S 的未满足集合中
7  return S
```

2.4 变异

我们对解进行一次局部搜索进行改进。这里我们选择了VRP问题中常用的几种邻域。

- 1. 删除一个请求并重新插入
 - 2. 交换两个请求
 - 3. 删除某一次回路并重新插入
- 局部搜索使得杂交出来的解得到进一步的改进。

2.5 选择

根据解的目标值即满足的请求数量及总共的费用成本来表示解的适应度。为了增加种群多样性,选择的同时我们令种群中不得出现适应度过于接近的解,只允许保留一个。

3 实验结果

为了验证算法的有效性,我们将本算法与多次迭代的局部搜索算法进行对比,算法迭代次数为500次。实验数据来源于实际数据,结果如下图:

表 1 未满足请求数量

遗传算法	4
局部搜索	4

表 2 路程费用

迭代次数	50	100	150	200	250	300	350	400	450	500
局部搜索	3129	3078	3068	3058	3058	3058	3051	3051	3034	3034
遗传算法	3051	3009	2987	2979	2974	2965	2965	2965	2964	2964

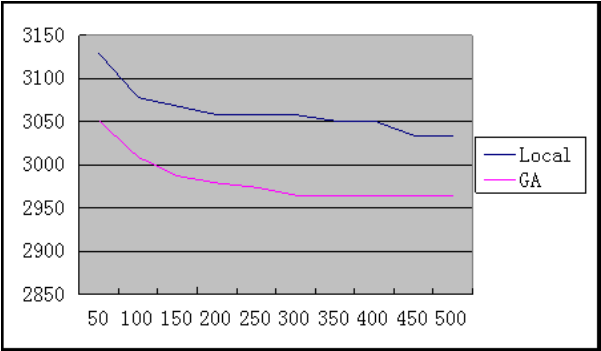


图 2 路程费用

表 3 算法 CPU 时间

迭代次数	50	100	150	200	250	300	350	400	450	500
局部搜索	320.45	637.15	955.8	1272.5	1591.1	1896.1	2213.1	2506.2	2823.0	3147.1
遗传算法	299.32	499.07	677.57	833.53	1003.2	1162.3	1322.9	1486.7	1642.0	1792.6

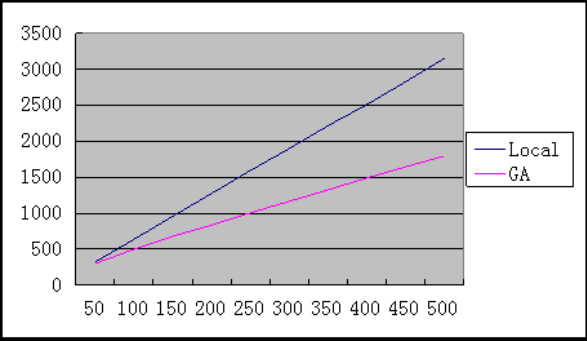


图 3 算法 CPU 时间

由上述图表可以得出,遗传算法在解的质量上远超过了局部搜索算法。算法时间上由于遗传算法的优势,随着种群不断进化,直接杂交得到的解大部分均已接近局部最优,从而节约大量计算时间。

4 结束语

VRP 问题随着与实际问题结合的更加紧密而出现了更多的变化,而这些变化恰好又是解决问题的关键所在。该文在描述问题的基础之上,提出了一种基于遗传算法并与实际结合的启发式算法,并用实验证明了算法的有效性。

参考文献:

[1] Christian Prins.A simple and effective evolutionary algorithm for the vehicle routing problem[J].Computers & Operations Research,2004, 31:1985-2002.

[2] Christian Prins.Two memetic algorithms for heterogeneous fleet vehicle routing problems[J].Engineering Applications of Artificial Intelligence,2009,22:916-928.

[3] Gendreau M,Laporte G,Seguin R.Stochastic vehicle routing[J].European Journal of Operational Research,1996,88:3-12.

[4] 吴璟莉,李陶深.有时间窗物流配送车路由问题的改进遗传算法[J].小型微型计算机系统,2004(10).

[5] 王沛栋,唐功友,李扬.带容量约束车辆路由问题的改进蚁群算法[J].控制与决策,2011(4).